**Fonctionnement du protocole DNSSEC**

**Authenticité et intégrité**

Afin d'assurer l'authenticité et l'intégrité des réponses, DNSSEC est basé sur un modèle de cryptographie à clé publique. Un serveur d'autorité calcule un hash de la réponse puis le signe avec une clé privée (secrète) avant d'envoyer le paquet et son hash au résolveur. Celui-ci sera à même de vérifier l'authenticité et l'intégrité des données en vérifiant que le hash déchiffré à l'aide de la clé publique associée à la clé privée correspond bien au hash recalculé de la réponse. La mise en place de ce nouveau protocole doit garantir le fonctionnement des demandes de résolutions classiques. de ce fait, on retrouvera les enregistrements du protocole DNS auxquels seront ajoutés de nouveaux enregistrement qui sont DNSKEY, RRSIG, NSEC, DS (RFC4034) et NSEC3, NSEC3PARAM.

**Les enregistrements liés aux signatures dans la zone.**

Nous nous baserons sur un fichier de zone existant dans une structure DNS classique. Soit la zone esit.be dont voici la configuration de base:

$TTL 86400

$ORIGIN esit.be.

@ 10D IN SOA ns1.esit.be. esit.it-xperts.be. (

2015010103 ; SERIAL

5H ; REFRESH

3600; RETRY

1W ; EXPIRE

1H ; TTL NEGATIF

)

@ IN NS ns1.esit.be.

@ IN NS ns1.belnet.be.

@ IN NS ns2.belnet.be.

@ IN A 193.191.131.25

@ IN AAAA 2001:06a8:3480:2::26

ns1 IN A 193.191.131.25

ns1 IN AAAA 2001:06a8:3480:2::25

www IN A 193.191.131.25

www IN AAAA 2001:06a8:3480:2::26

Les enregistrements vont être regroupés en fonction du nom et du type pour former des RRSET (Ressources Record Set). Prenons dans le fichier précédent les enregistrements de type NS associés à esit.be. Cette notion sera importante par la suite car la signature dans DNSSEC porte sur un RRSET et non pas sur un enregistrement individuel.

|  |  |
| --- | --- |
| @ IN NS  @ IN NS  @ IN NS | ns1.esit.be.  ns1.belnet.be.  ns2.belnet.be. |

**L' enregistrement DNSKEY**

L'enregistrement DNSKEY est utilisé pour transmettre une clé publique entre le résolveur et le serveur de nom. Cette clé publique est celle qui est associée à la clé privée avec laquelle le serveur d'autorité va signer les hashs des enregistrements RRSET . Le résolveur utilisera donc la clé publique écrite dans l'enregistrement DNSKEY pour vérifier la signature du serveur d'autorité, s'assurant ainsi de l'intégrité et de l'authenticité du message. Il est à noter qu'une clé publique est associée à une zone et non à un serveur. Nous retrouverons deux types de clefs qui sont KSK et ZSK. La clef KSK signera seulement les DNSKEY RRSET tandis que la clef ZSK signera les autres RRSET gans la zone.

Nous analyserons la structure des enregistrements présents dans la zone signée de esit.be

@ **86400 DNSKEY 256 3 8** (

AwEAAb8sqNVtoIzysSl3A25LSwE4C3fVkiHx

P9/FWtZx+DWzhIcOHAQ+paNLsBOHSEj7Fd/G

AS4l6DBDQxB992H7qDIkq+o5alUqn9VFPWOh

m7fDx2HwlkYCPOqQSC+IMqZCudePLuvc/zXu

BhUZQGRNdW/2f35dr5HHKrb1h5ORfIgt

) ; **key id = 6228**

86400 correspond au TTL de l'enregistrement DNSKEY

256 correspond au champ flags. (256 pour ZSK et 257 pour KSK)

3 correspond au protocole. Il aura tout le temps la valeur 3 pour un enregistrement DNSKEY.

8 correspond au champ algorithme. Les valeurs suivantes pourront être utilisées:

0 – reserved

1 – RSA/MD5 (deprecated)

2 – Diffie/Hellman

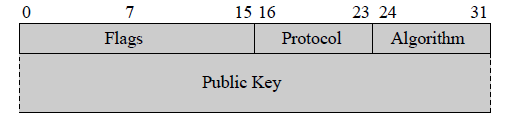
3 – DSA/SHA-1 (optional)

4 – reserved

5 – RSA/SHA-1 (mandatory in validator)

8 – RSA/SHA-256 (recommended)

10-RSA/SHA-512 (recommended)

****

**L'enregistrement RRSIG (Resource record SIGnature)**

L'enregistrement RRSIG contient la signature de l'enregistrement envoyé par le serveur d'autorité. Il s'agit de la signature obtenue en signant le hash du RRset avec la clé privée du serveur d'autorité, c'est-à-dire à celle contenue dans l'enregistrement DNSKEY. C'est cette signature que le résolveur vérifiera par la suite. Ainsi, il existera un enregistrement RRSIG pour chaque enregistrement de la zone dans le fichier de zone signé.

86400 A 193.191.131.25

86400 RRSIG **A 8 2 86400 20140714102050** (

**20140614092050 6228 esit.be.**

SCbnsSfMIhZGtF7A8JYlr93yP42wC/r8z2uZ

QyDjj6tufw94jDqiia/EwYSpgStsBkDzYC0v

2GmE6DFnjvSHB+H0e3H3/fCmmlEcGuFvscgc

AsIF0xg9HrQCWrPfdA4gdiEzv53tx9aVqRsY

EMgUWRgqNqcUNxDV3jPwZY9SrEQ= )

A: Le champ 'type covered' correspond au type d'enregistrement qui a été signé. Dans notre cas, il s'agit d'un RRSET couvrant le type A

8: Algorithme utilisé (voir descriptif de DNSSEK)

2: Le champ 'label' indique le niveau de hiérarchie dans le nom contenu dans le RRSET signé. Dans notre cas, esit.be contient deux champs (be et esit), nous aurons la valeur 2. Dans la même zone, pour un autre RRSIG associé au nom ns1.esit.be, nous aurons la valeur 3 ( be, esit et ns1)

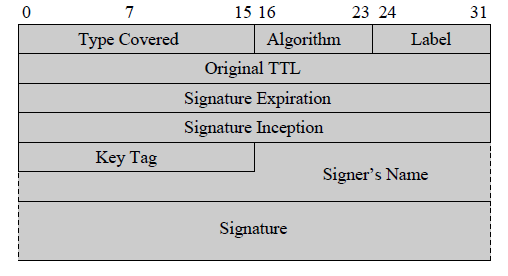
86400: Le TTL d'origine

20140714102050: date d'expiration de la signature

20140614092050: date de la signature

6228: Identifiant de la clef utilisée. Nous retrouvons cette clef dans l'enregistrement DNSKEY décrit dans la paragraphe précédent.

esit.be.: Indique le nom de la zone détentrice de la clé utilisée pour la signature. Ce nom est utile pour la vérification de la signature.



**Les enregistrements liés à la preuve de non existence.**

**Introduction**

Assurer l'authenticité et l'intégrité des réponses est possible tant qu'une réponse peut être fournie. On peut se poser la question de savoir ce qui se passe lorsque la requête envoyée vers le serveur DNS ne correspond à aucune réponse. C'est la raison d'exister des enregistrements NSEC et NSEC3.

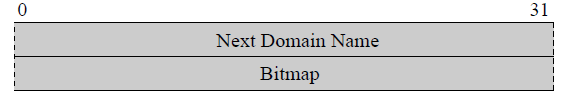
**Enregistrement NSEC**

Un enregistrement NSEC contient les informations nécessaires à l’identification des enregistrements

existants pour un nom donné, ainsi que le prochain nom existant dans la zone (dans l’ordre lexicographique). Ces deux informations suffisent pour prouver qu’un enregistrement ou

qu’un domaine n’existe pas.

Un enregistrement NSEC a la structure suivante:



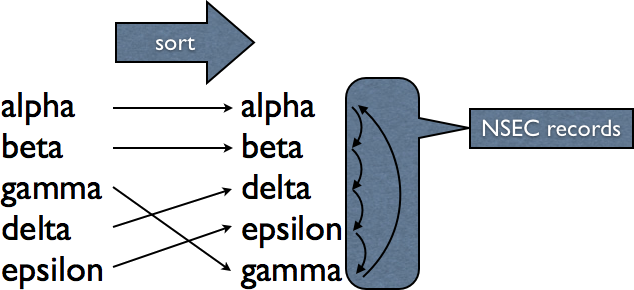
**ex:**

ns1.esit.be. 3600 NSEC **www.esit.be**. **A AAAA RRSIG NSEC**

www.esit.be. correspond au nom de domaine suivant

A AAAA RRSIG NSEC correspond à une liste de RSET correspondant au domaine suivant

Le NSEC du dernier nom de domaine pointe vers le premier nom de domaine dans la liste.



Un des problèmes de ce type d'enregistrement est la possibilité, sans que les transferts ne soient validés pour une zone, de pouvoir récupérer l'ensemble des différents noms de domaine. Pour montrer cette faille, nous allons utiliser l'utilitaire de linux ldns-walk (package ldnsutils). En interrogeant le serveur local, nous obtiendrons les informations suivantes:

MyServer#ldns-walk @127.0.0.1 esit.be

esit.be. esit.be. A NS SOA AAAA RRSIG NSEC DNSKEY

ns1.esit.be. A AAAA RRSIG NSEC

www.esit.be. A AAAA RRSIG NSEC

Pour éviter ce problème, deux autres types d'enregistrements sont apparus: NSEC3PARAM et NSEC3

**Enregistrement NSEC3**

Cet enregistrement permet la gestion des réponses négatives tout en luttant contre des attaques de type "énumération de zone" (voir section 1.5). Pour ce faire, l'enregistrement NSEC3 utilise une fonction de hachage appliquée aux noms renvoyés. Ainsi, sera envoyé le hash du prochain nom de la zone, ne permettant alors pas de reconstruire la zone entière.

NSEC3PARAM est utilisé par le serveur DNS pour calculer et déterminer quels sont les enregistrement NSEC3 à inclure pour les réponses de non existence.

Cet enregistrement comprendra les champs suivants:

- Algorithme de hashage

- Flag "Opt-out" indiquant si les délégations sont signées ou pas

- Itérations: combien de fois l'algorithme de hashage doit-il être appliqué

-Salt: valeur de Salt pour le calcul du hashage

0 NSEC3PARAM 1 0 100 85A9F28CC4264523

Dans notre cas de figure:

1: correspond à SHA-1

0: Opt-out

100: itérations

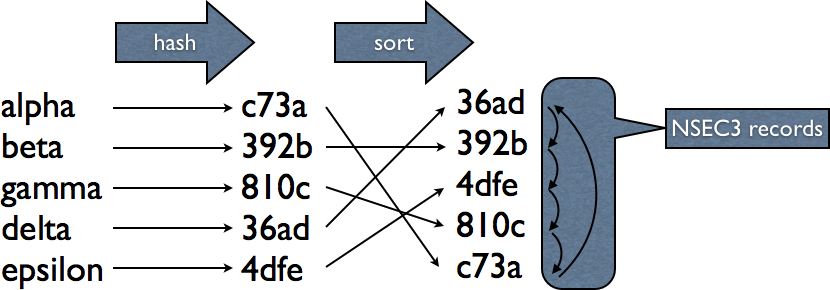
85A9F28CC4264523: Salt

Pour les enregistrements de type NSEC3, nous aurons la configuration suivante:

**C0O2TRHFHTFE2HOV1R6P1GGOJ58J5V56**.esit.be. 3600 IN **NSEC3 1 0 100 85A9F28CC4264523** **Q3D1L6GDFG9KAVMVPQANF1N9VI6G1C60** A NS SOA AAAA RRSIG DNSKEY NSEC3PARAM

La partie en rouge correspond bien aux données reprises dans l'enregistrement NSEC3PARAM. Nous remarquons que les noms de domaine ont été hashés et de ce fait, lorsque le résolveur voudra vérifier la réponse négative, il va générer le hash du nom demandé, soit h2 et vérifier que h2 est compris entre h1 et h3 (h1 et h3 correspondent aux valeurs en bleu dans la copie d'écran).

A ce niveau, il est conseillé de remplacer régulièrement le Salt.

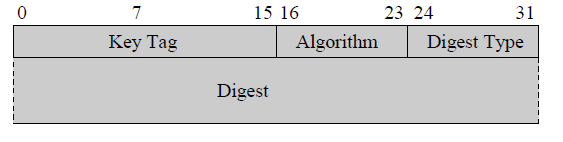


**Enregistrement DS (Delegation Signer)**

L'enregistrement DS assure la chaîne de confiance dans les liaisons entre les serveurs DNS parents et enfants. L'enregistrement DS sera placé dans la zone parent et sera lié à l'enregistrement DNSKEY de la zone enfant. Cet enregistrement DS contiendra un hash de la clef KSK (DNSKEY) de la zone enfant, stockée avec l'enregistrement NS indiquant la délégation vers la zone enfant. Cet enregistrement DS est signé avec le reste des données de la zone parente.

Cet enregistrement est demandé par le résolveur au serveur autoritaire dès qu'il a besoin de vérifier une signature reçue par la zone fille. Ainsi, la zone mère lui indique qu'il peut faire confiance à ce serveur en lui renvoyant un hash de la clé signé par elle-même. Le résolveur sait ainsi qu'il peut lui faire confiance. À chaque maillon de la chaîne, nous retrouvons le même schéma : un enregistrement DS permet d’avoir confiance en une KSK, une KSK permet d’avoir confiance en l’ensemble des clés de la zone (KSK et ZSK) et les ZSK permettent alors d’avoir confiance dans les autres enregistrements.

Un enregistrement DS a la structure suivante:

****

**Champ Key Tag, Algorithm:** Idem RRSIG

**Champ Digest Type:** Fonction de hachage cryptographique utilisée pour construire le hash situé dans le champ Digest. On utilisera en général SHA-1.

**Champ Digest:** *Digest* = *Hash*(*nomfille* | *DNSKEY*) | représente une concaténation

**Exemple:**

esit.be. IN DS **42235** **8** **1** 4C2DB53EE7E63B97C8A89F96E7ADF5D297FE1AEA

esit.be. IN DS 42235 8 **2** F0C9B4634A04D65F89D69A3340DD1D4BBA4EB2E80A457FD983FE808B 76149F1F

42235:the key ID (ID de la clef DNSKEY correspondant à la KSK dans la zone enfant.

8: numéro de l'algorithme utilisé pour DNSKEY (RSA/SHA-256)

1: sha-1

2: sha-256

**Zone Signing Key (ZSK)**

Comme renseigné dans la section traitant de l'enregistrement DNSKEY, la Zone Signing Key (ZSK) correspond à la clé privée utilisée pour signer une zone, c'est-à-dire les enregistrements demandés par les serveurs récursifs. La clé publique correspondante est connue des serveurs récursifs afin qu'ils puissent vérifier les hash des réponses. Une clef n'expire jamais. Seules les signatures RRSIG ont une période de validité et l'on pourrait se contenter de signer les enregistrements avec la même clef. Malgré tout, pour des questions de sécurité, on conseille de renouveler la clef tous les mois. Les serveurs DNSSEC intègre un mécanisme de rotation de clef de sorte que chaque mois, la clef soit changée.

Une première technique consiste à générer une nouvelle clef et à signer la zone avec cette clef en laissant l'ancienne clef présente dans le fichier de zone le temps d'expiration des données dans les caches. Cette technique est pénalisante du fait de la présence d'une double signature pour chaque enregistrement.

Une autre technique consiste à pré publier la nouvelle clef. Nous retrouverons une clef active et une clef passive dans le sens où une seule des deux est utilisée pour signer les enregistrements de la zone, l'autre n'étant pas utilisée. On peut ajouter la nouvelle clef au moins juste avant la valeur TTL de la clef voir au plus tôt les deux clefs en même temps au début de la période du mois.

En fin de période du mois écoulée, la zone est signée avec la clef passive, elle même signée avec la clef KSK.

**Key Signing Key (KSK)**

La Key Signing Key est une clé privée destinée à signer une clé de zone (ZSK). Une KSK n'est utilisée que pour signer un enregistrement DNSKEY. La clé KSK fait office de maillon de confiance dans la chaîne de confiance décrite en section. Les signatures des clefs ZSK ont, en général, une durée de validité de 1 an. Le remplacement de cette clef est plus complexe que les clefs ZSK du fait que l'on doit également intervenir dans la zone parent.

1- Nous ne pouvons contrôler quand le parent changera ou publiera le nouvel enregistrement DS.

2- Nous devons maintenir l'ancienne KSK le temps que le TTL de l'enregistrement DS soit arrivé à expiration et ce pour les problèmes des caches.

Il faudra également tenir compte du fait que le parent puisse accepter ou pas plusieurs enregistrement DS pour une même zone enfant. Nous allons envisager les deux cas de figure.

*Pour les parents n'acceptant qu'un seul enregistrement DS:*

1. Introduire une nouvelle clef KSK dans la zone enfant.
2. Signer les clefs ZSK avec la nouvelle clef KSK en maintenant les anciennes signatures.
3. Attendre que les caches aient la nouvelle version de la zone avec les nouvelles signatures.
4. Publier le nouvel enregistrement DS chez le parent, qui remplacera alors l'ancien.
5. Attendre que le parent publie ce nouvel enregistrement DS.
6. Attendre le plus long temps entre le TTL de l'ancien et du nouveau DS.
7. Déprécier l'ancienne clef KSK et supprimer les anciennes signatures des clefs ZSK.

*Pour les parents acceptant plusieurs enregistrements DS:*

1. Introduire une nouvelle clef KSK dans la zone enfant.
2. Signer les clefs ZSK avec la nouvelle clef KSK en maintenant les anciennes signatures.
3. Publier le nouvel enregistrement DS en maintenant l'ancien DS.
4. Attendre que le parent publie ce nouvel enregistrement DS.
5. Attendre le plus long temps entre le TTL de l'ancien et du nouveau DS.
6. Déprécier l'ancienne clef KSK et supprimer les anciennes signatures des clefs ZSK.
7. Supprimer l'ancien enregistrement DS du parent.

Le calcul de la signature est donné par la formule suivante :

*signature* = *KZSK*(*Hash*(*RRSIGDATA* | *RR*(1) | *RR*(2) | ...))

où RRSIGDATA est la concaténation de tous les champs de l'enregistrement RRSIG sauf le champ Signature, et

*RR*(*i*) = *nomdetenteur* | *type* | *classe* | *TTL* | *TailleRDATA* | *RDATA*,

où *nomdetenteur* est le nom du signataire en forme canonique, et tel que tous les RR(i) vérifient

le nom du détenteur correspond au champ Signer's Name du RRSIG

la classe est la même que celle du RRSIG

le type fait partie des types listés dans le champ Type Covered Field

le TTL est le même que clui du RRSIG

**Le concept de Trust Anchor**

Tout paraît simple dans l'application de cette chaine de confiance, mais c'est l'un des éléments le plus critique de DNSSEC. Pour que celle-ci puisse être appliquée, il faut introduire un nouveau concept, celui du **trust anchor**. En effet, pour faire confiance au fils, on doit demander au père. Mais quid du serveur racine ?

On appelle *trust anchor* (encre de confiance) la première clé de la chaine a qui on fait confiance. Cette clé est la clé publique associée à la KSK de la racine. Tous les serveurs sont censés connaître cette clé qui est entrée manuellement.

Cette trust anchor est un élément critique de la chaine de confiance puisqu'il suffit qu'un pirate réussisse à trouver la clé privée pour que toute la chaine de confiance soit compromise.

**Exemple de mise en place d'un ilot sécurisé**

DNSSEC est à ce jour (jeudi 9 juin 2011) en cours de déploiement, et encore de nombreux serveurs n'ont pas implémenté cette fonctionnalité (voir section [2](http://ensiwiki.ensimag.fr/index.php/Introduction_%C3%A0_DNSSEC#D.C3.A9ploiement_.C3.A0_l.27heure_actuelle)). Il existe cependant des ilots sécurisés, qui ne peuvent pas faire valider leurs clés par leurs serveurs autoritaires respectifs, mais qui ont quand même souhaités développer des échanges sécurisés pour leurs sous-domaines.

Pour mettre en place cela, on doit introduire manuellement la clé publique associée à la KSK de la zone mère dans les différents autres serveurs. De plus, pour pouvoir indiquer qu'une zone fille est sécurisée, il est nécessaire d'introduire l'enregistrement DS de la fille à l'intérieur du fichier de zone autoritaire manuellement aussi.

**DNSSEC Lookaside Validation (DLV)**

Une alternative à la gestion manuelle des trust anchors est d'utiliser DLV (DNSSEC Lookaside Validation), normalisé plus tard. L'idée est de dissocier la racine du DNS et la racine de signature DNSSEC (pour des problèmes politiques). Avec DLV, et contrairement au DNSSEC, plusieurs "racines" peuvent signer un même domaine (une racine DLV qui signe .org et une autre qui signe site.org). Le résolveur DLV peut utiliser l'algorithme qu'il veut. Cette alternative a eu beaucoup de détracteurs, car elle aurait retardé la signature de la racine. Maintenant que la racine est signée, la clé de la racine devrait suffire en tant que trust anchor. Cela explique aussi le fait qu'il y ait très peu de TLD qui ont leur trust anchor publiée dans le dépot DLV de l'ISC (Internet Systems Consortium).

**Remarques importantes**

DNSSEC peut générer des problèmes de compatibilité avec les équipements réseau qui prennent en charge DNS. Les pare-feu, routeurs et autres matériels réseau doivent être compatibles avec DNSSEC. Le problème est que les paquets DNSSEC sont plus volumineux que les paquets DNS, limités à 512 octets. De plus, en raison des limites de taille de MTU, le protocole UDP ne peut pas toujours prendre en charge les paquets DNSSEC. Utiliser TCP alourdit considérablement le trafic. Certains périphériques ne prennent même pas en charge DNS sur TCP. Il y a donc une longue phase de tests au niveau matériel pour s'assurer du bon fonctionnement de DNSSEC.

**Passage à DNSSEC pour BIND9**

Nous partirons du principe que votre BIND9 est correctement configuré pour le DNS conventionnel avec votre zone existante.

Première étape: activer DNSSEC sur votre serveur BIND9.

Cette étape consiste à apporter les modifications suivantes dans le fichier de configuration de votre serveur.

dnssec-validation auto;

dnssec-enable yes;

dnssec-lookaside auto;

Une fois le dnssec activé sur le serveur, nous pouvons signer les fichiers de zone. Pour ce faire, nous allons utiliser l'utilitaire zonesigner. Si cet outil n'est pas présent dans votre distribution, vous devrez installer la package dnssec-tools.

Deuxième étape: signer votre fichier de zone avec zonesigner.

zonesigner -genkeys -usensec3 -zone esit.be esit.be.zone

L'option -genkeys permet de générer les clefs (KSK et deux clefs ZSK) et de signer la zone esit.be correspondant au fichier de zone esit.be.zone.

L'option -usensec3 permet d'utiliser les enregistrements NSEC3 au lieu des enregistrements NSEC.

Une fois la commande exécutée, nous obtiendrons les fichiers suivants:

-> Les deux clefs ZSK comprenant pour chacune d'elles la clef privée et la clef publique.

Kesit.be. .+008+06228.key

Kesit.be.+008+06228.private

Kesit.be.+008+17254.key

Kesit.be.+008+17254.private

-> La clef KSK comprenant la clef privée et la clef publique

Kesit.be.+008+31818.key

Kesit.be.+008+31818.private

-> Les enregistrements de type DS pour la zone parent

dsset-esit.be.

-> Fichier référençant les clefs de la zone esit.be

esit.be.krf

-> Fichier de zone signé.

esit.be.zone.signed

Troisième étape: une fois la zone signée, il ne reste plus qu'à la charger en modifiant la configuration de la zone au niveau du serveur bind en éditant le fichier /etc/bind/named.conf.local

zone "esit.be" IN {

type master;

**file "esit.be.zone";**

allow-transfer {...}

notify yes;

};

Nous remplacerons le nom du fichier d'origine "esit.be.zone" par "esit.be.zone.signed"

zone "esit.be" IN {

type master;

**file "esit.be.zone.signed";**

allow-transfer { 193.190.198.14; 193.190.182.40; 2001:6a8:3c80::14; 200$

notify yes;

};

Attention: le fichier de zone non signé doit toujours être conservé. C'est dans celui-ci que l'on apportera les modifications à la zone et qui sera à chaque fois signé de nouveau.

Quatrième étape: charger le record DS dans la zone parent. La procédure à suivre dépendra du provider chez lequel vous êtes.

Rappelons nous plusieurs points importants dans la gestion de DNSSEC:

1- La rotation automatique des clefs ZSK chaque mois et le remplacement manuel de la clef KSK chaque année. Si vous passez au delà de la période de validité des signatures, votre zone ne sera plus accessible te les demandes de résolution fourniront l'erreur SERVFAIL.

Un outil fourni avec dnssec-tools est roolerd. Le fichier de ce deamon peut être réalisé au moyen de l'utilitaire rollinit. Vous pouvez également éditer le fichier manuellement /etc/dnssec-tools/dnssec-tools.rollrec. Un exemple d'utilisation de la commande rollint:

rollinit -zonefile /var/cache/bind/esit.be.zone.signed -keyrec /etc/bind/esit.be.krf -admin dnsadmin@esit.be esit.be >>/etc/dnssec-tools/dnssec-tools.rollrec

2-La modification du fichier de zone. Toute modification doit être apportée dans le fichier non signé. Nous utiliserons la commande zonesigner avec la syntaxe suivante:

zonesigner -zone esit.be esit.be.zone esit.be.zone.signed